IN THE CLAIMS

Please amend the claims as follows:

1. (Original) An apparatus comprising:

a load/store unit that includes a retry logic that is to retry access to a resource after receipt of a negative acknowledgement for an attempt to access the resource by the load/store unit; and

a congestion detection logic to output a signal that indicates that the resource is congested based on receipt of a consecutive number of negative acknowledgments in response to access requests to the resource.

2. (Original) The apparatus of claim 1 further comprising a congestion control logic to disable the retry logic from retry accesses to the resource based on receipt of the signal from the congestion detection logic.

3. (Original) The apparatus of claim 2, wherein the congestion control logic is to exponentially increase the delay after the congestion detection logic is to detect congestion while the resource is currently congested.

4. (Original) The apparatus of claim 2, wherein the congestion control logic is to exponentially decrease the delay after the congestion detection logic receive a number of positive acknowledgements in response to access requests to the resource.

5. (Original) A processor comprising:

a functional unit to attempt to access data from memory coupled to the processor based on an access request, wherein the functional unit is to retry attempts to access of the data based on other access requests after receipt of a negative acknowledgement in response to the attempt to access the data; and

a congestion detection logic to detect congestion of access of the data based on receipt of a consecutive number of negative acknowledgments that exceed a threshold prior to access of the data; and

a congestion control logic to disable the functional unit from the attempts to access the data for a time period after congestion is detected.

6. (Original) The processor of claim 5, wherein the congestion control logic is to exponentially increase the time period after the congestion detection logic is to detect congestion while access to other data in the memory is congested.

7. (Original) The processor of claim 6, wherein the congestion control logic is to exponentially decrease the time period after the congestion detection logic receives a number of positive acknowledgements in response to attempts to access data in the memory.

8. (Original) A processor comprising:

a functional unit to attempt to access a cache line in a cache memory coupled to the processor based on an access request, wherein the functional unit is to retry attempts to access the cache line based on additional access requests after receipt of a negative acknowledgement in response to the attempt to access the data;

a congestion detection logic to detect congestion of access of the cache line based on an average number of negative acknowledgments received that exceed a threshold prior to access of the data; and

a congestion control logic to disable the functional unit from attempts to access the cache line for a time period after congestion is detected.

9. (Original) The processor of claim 8, wherein the average number of negative acknowledgements is within a window and wherein the congestion detection logic is to move the window over time of attempts to access the cache line by the functional unit.

10. (Original) The processor of claim 8, wherein the congestion control logic is to exponentially increase the time period after the congestion detection logic is to detect congestion while access of other cache lines in the cache memory is congested.

11. (Original) The processor of claim 8, wherein the congestion control logic is to exponentially decrease the time period after the congestion detection logic receives a number of positive acknowledgements in response to attempts to access other cache lines in the cache memory.

12. (Original) A system comprising:

a cache memory to store data; and

a first processor to attempt to access the data from the cache memory based on access requests, wherein the first processor includes a congestion detection logic to detect congestion of access to the data based on receipt of a consecutive number of negative acknowledgements in response to the access requests.

13. (Original) The system of claim 12 further comprising:

a second processor associated with the cache memory;

a hub controller to receive the access requests from the first processor, the hub controller to forward the access requests to the second processor, wherein the second processor is to determine whether the data in the cache memory is accessible.

14. (Original) The system of claim 13, wherein the second processor is to transmit a negative acknowledgement back to the first processor through the hub controller if the data is not accessible, the second processor to transmit a positive acknowledgment back to the first processor through the hub controller if the data is accessible.

15. (Original) The system of claim 12, wherein the first processor further comprises a congestion control logic to disable the first processor from transmitting the access requests if the congestion detection logic determines that access to the data is congested.

16. (Original) The system of claim 12, wherein the congestion control logic is to disable the first processor from transmitting the access requests for a time period, wherein the time period is based on an exponential back off delay operation.

17. (Original) A system comprising:

a resource; and

a first processor having a load/store functional unit, the load/store functional unit to attempt to access the resource based on access requests, wherein the first processor includes a congestion detection logic to detect congestion of access of the resource based on a consecutive number of negative acknowledgements received in response to the access requests prior to receipt of a positive acknowledgment in response to one of the access requests within a first time period.

18. (Original) The system of claim 17 further comprising:

a second processor associated with the resource;

a hub controller to receive the access requests from the first processor, the hub controller to forward the access requests to the second processor, wherein the second processor is to determine whether the resource is accessible.

19. (Original) The system of claim 18, wherein the second processor is to transmit a negative acknowledgement back to the first processor through the hub controller if the resource is not accessible, the second processor to transmit a positive acknowledgment back to the first processor through the hub controller if the resource is accessible.

20. (Original) The system of claim 17, wherein the first processor further comprises a congestion control logic to disable the load/store functional unit from attempting to access the resource if the congestion detection logic is to detect congestion of access of the resource

21. (Original) The system of claim 17, wherein the congestion control logic is to disable the load/store unit from attempts to access the resource for a second time period, wherein the second time period is based on an exponential back off delay operation.

22. (Original) A system comprising:

a cache memory to include a number of cache lines for storage of data; and

at least two processors, wherein a first processor of the at least two processors is to attempt to access the data in one of the number of cache lines based on access requests, wherein the first processor includes a congestion detection logic to detect congestion of access of a first cache line of the number of cache lines based on a ratio of a number of negative acknowledgments to a number of positive acknowledgments received in response to the access requests.

23. (Original) The system of claim 22, wherein a second processor of the at least two processors is associated with the cache memory and wherein the system further comprises a hub controller, the hub controller to receive the access requests from the first processor, the hub controller to forward the access requests to the second processor, wherein the second processor is to determine whether the one of the number of cache lines is accessible.

24. (Original) The system of claim 23, wherein the second processor is to transmit a negative acknowledgement back to the first processor through the hub controller if the one of the number of cache lines is not accessible, the second processor to transmit a positive acknowledgement back to the first processor through the hub controller if the one of the number of cache lines is accessible.

25. (Original) The system of claim 22, wherein the first processor further comprises a congestion control logic to disable, for a time period, the first processor to attempt to access the data if the congestion detection logic is to detect congestion of access of the first cache line.

26. (Original) The system of claim 25, wherein the congestion control logic is to exponentially increase the time period after the congestion detection logic is to detect congestion while access to other cache lines in the cache memory.

27. (Currently amended) A method comprising:
transmitting access requests, by a first processor, to access data in a memory;

receiving, by the first processor, a positive acknowledgement or a negative acknowledgment from a second processor that is associated with the memory based on one of the number of access requests; and

detecting congestion of the data based on receipt, by the first processor, of a consecutive number of negative acknowledgements that exceed a first threshold, prior to receipt, by the first processor, of a positive acknowledgment.

28. (Original) The method of claim 27 further comprising controlling access to the data in the memory if the consecutive number of negative acknowledgements, received by the first processor, exceeds the first threshold, prior to receipt of the positive acknowledgement.

29. (Original) The method of claim 28, wherein controlling access to the data in the memory comprises disabling transmitting of the access requests, by the first processor, for a time period.

30. (Original) The method of claim 29, wherein controlling access to the resource comprises exponentially increasing the time period upon determining that the congestion is detected for other data in the memory while the time period has not expired.

31. (Original) A method comprising:

accessing, by at least one processor, a resource based on an access request;

receiving a positive acknowledgement if the resource is accessible;

receiving a negative acknowledgement if the resource is not accessible;

retrying accessing, by the at least one processor, of the resource based on a number of access requests; and

detecting that a consecutive number of negative acknowledgements exceeds a first threshold within a time period, prior to receiving a positive acknowledgments.

32. (Original) The method of claim 31 further comprising controlling access to the resource if the consecutive number of negative acknowledgements, received by the at least one processor, exceeds the first threshold, prior to receipt of the positive acknowledgement.

33. (Original) The method of claim 31, wherein controlling access to the resource comprises disabling transmitting of the access requests, by the first processor, for a time period.

34. (Currently amended) A computer storage ~~machine-readable~~ medium that provides instructions, which when executed by a machine, cause said machine to perform operations comprising:

transmitting access requests, by a first processor, to access data in a memory;

receiving, by the first processor, a positive acknowledgement or a negative acknowledgment from a second processor that is associated with the memory based on one of the number of access requests; and

detecting congestion of the data based on receipt, by the first processor, of a consecutive number of negative acknowledgements that exceed a first threshold, prior to receipt, by the first processor, of a positive acknowledgment.

35. (Currently amended) The computer storage ~~machine-readable~~ medium of claim 34 further comprising controlling access to the data in the memory if the consecutive number of negative acknowledgements, received by the first processor, exceeds the first threshold, prior to receipt of the positive acknowledgement.

36. (Currently amended) The computer storage ~~machine-readable~~ medium of claim 35, wherein controlling access to the data in the memory comprises disabling transmitting of the access requests, by the first processor, for a time period.

37. (Currently amended) The computer storage ~~machine-readable~~ medium of claim 36, wherein controlling access to the resource comprises exponentially increasing the time period upon determining that the congestion is detected for other data in the memory while the time period has not expired.

38. (Currently amended) A computer storage ~~machine-readable~~ medium that provides instructions, which when executed by a machine, cause said machine to perform operations comprising:

accessing, by at least one processor, a resource based on an access request;

receiving a positive acknowledgement if the resource is accessible;

receiving a negative acknowledgement if the resource is not accessible;

retrying accessing, by the at least one processor, of the resource based on a number of access requests; and

detecting that a consecutive number of negative acknowledgements exceeds a first threshold within a time period, prior to receiving a positive acknowledgments.

39. (Currently amended) The computer storage ~~machine-readable~~ medium of claim 38 further comprising controlling access to the resource if the consecutive number of negative acknowledgements, received by the at least one processor, exceeds the first threshold, prior to receipt of the positive acknowledgement.

40. (Currently amended) The computer storage ~~machine-readable~~ medium of claim 39, wherein controlling access to the resource comprises disabling transmitting of the access requests, by the first processor, for a time period.